APPENDIX A

Oracle7 Parallel Server Concepts and Administrator's Guide

Parallel Hardware Architecture

The parallel database server can use various machine architectures which allow parallel processing. This chapter describes the range of available hardware implementations and surveys their advantages and disadvantages.

Overview

This section covers the following topics:

Oracle configurations support parallel processing within a machine, between machines, and between nodes. There is no advantage to running Oracle Parallel Server on a single node and a single system image--you would incur overhead and receive no benefit. With standard Oracle you do not have to do anything special on shared memory configurations to take advantage of some parallel processing capabilities.

Although this manual focuses on Oracle Parallel Server with shared nothing/shared disk architecture, the application design issues discussed in this book may also be relevant to standard Oracle systems.

Parallel Processing Hardware Implementations

Parallel processing hardware implementations are often categorized according to the particular resources which are shared. The following categories are described in this chapter:

- shared memory systems
- shared disk systems
- shared nothing systems

These implementations can also be described as "tightly coupled" or "loosely coupled," according to the way in which communication between nodes is accomplished:

Implementation Type	Tightly Coupled	Loosely Coupled
Shared Memory	SMP	NUMA
Shared Disk	MPP	Clusters
Shared Nothing	MPP	Clusters

Table 3 - 1. Classification of Parallel Processing Hardware Implementations

Attention: Oracle supports *all* these different implementations of parallel processing.

The following table shows interrelations between various hardware architectures and available Oracle options.

System Configuration	Shared Memory	Shared Disk	Shared Nothing
Type of Oracle	standard Oracle	Oracle Parallel Server	Oracle Parallel Server
Parallel Query Option	available	available	available

Table 3 - 2. Hardware Architecture and Oracle Solutions

Note: Support for any given Oracle configuration is platform-dependent; check to confirm that your platform supports the configuration you want.

The preceding table assumes that in a shared nothing system the software enables a node to access a disk from another node. For example, the IBM SP2 features a virtual shared disk: the disk is shared through software.

Application Profiles

Online transaction processing (OLTP) applications tend to perform best on symmetric multiprocessors or clusters. Decision support (DSS) applications tend to perform well on massively parallel systems. Application profiles and parallel processing hardware implementations can typically be represented as follows:

Implementation Type	OLTP	DSS
Shared Memory (SMP)	Good	Good
Shared Disk (Cluster)	OK if carefully implemented	Good
Shared Nothing (MPP)	Not optimal	Good

Table 3 - 3. Parallel Processing Implementations and Application Profiles

Choose the implementation that provides the power you need for the application(s) you require.

Required Hardware and Operating System Software

Each hardware vendor implements parallel processing in its own way, but the following common elements are required for Oracle Parallel Server:

- High Speed Interconnect
- Globally Accessible Disk or Shared Disk Subsystem
- Distributed Lock Manager

High Speed Interconnect

This is a high bandwidth, low latency communication facility between the various nodes for lock manager and cluster manager traffic. The interconnect can be Ethernet, FDDI, or some other proprietary interconnect method. If the primary interconnect fails, a back-up interconnect is usually available. The back-up interconnect will ensure high availability, and prevent a single point of failure.

Globally Accessible Disk or Shared Disk Subsystem

All nodes in a loosely coupled or massively parallel system have simultaneous access to shared disks. This gives multiple instances of Oracle7 concurrent access to the same database. These shared disk subsystems are most often implemented via a shared SCSI or twintailed SCSI (common in UNIX) connected to a disk farm. On some MPP platforms, such as IBM SP, disks are associated to nodes and a virtual shared disk software layer enables global access to all nodes.

Distributed Lock Manager

The distributed lock manager (DLM) consists of software and hardware that coordinates resource sharing in a loosely coupled or massively parallel system. The distributed lock manager tracks "ownership" of a resource, accepts requests for resources from processes, notifies requesting processes when a resource is available, and grants shared or exclusive access to a resource for a process.

Oracle uses the DLM to coordinate modifications of data blocks, maintenance of cache consistency, recovery of failed nodes, transaction locks, dictionary locks, and SCN locks.

See Also: "Distributed Lock Manager: Access to Resources" ■.

Shared Memory Systems

This section describes:

- Tightly Coupled Systems
- Uniform and Non-Uniform Memory Access
- Summary: Shared Memory Systems

Tightly Coupled Systems

Tightly coupled shared memory systems, illustrated in Figure 3 - 1, have the following characteristics:

- Multiple CPUs share memory.
- Each CPU has full access to all shared memory through a common bus.

- Communication between nodes occurs via shared memory.
- Performance is limited by the bandwidth of the memory bus.

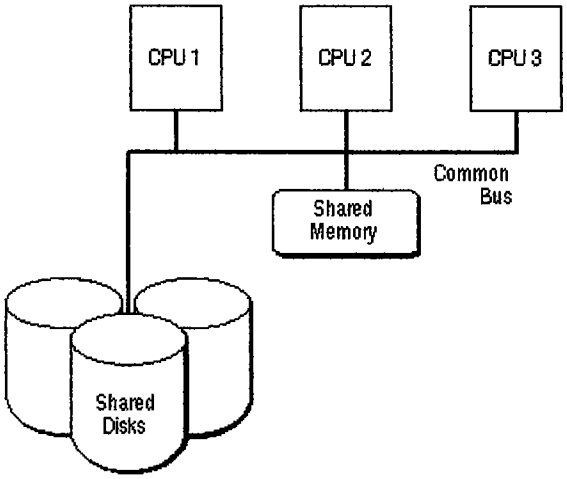

Figure 3 - 1. Tightly Coupled Shared Memory System

Symmetric multiprocessor (SMP) machines are often nodes in a cluster. Multiple SMP nodes can be used with Oracle Parallel Server in a tightly coupled system, where memory is shared among the multiple CPUs, and is accessible by all the CPUs through a memory bus. Examples of tightly coupled systems include the Pyramid, Sequent, and Sun SparcServer.

It does not make sense to run Oracle Parallel Server on a single SMP machine, because the system would incur a great deal of unnecessary overhead from DLM accesses.

Performance is potentially limited in a tightly coupled system by a number of factors. These include various system components such as the memory bandwidth, CPU to CPU communication bandwidth, the memory available on the system, the I/O bandwidth, and the bandwidth of the common bus.

Uniform and Non-Uniform Memory Access

Shared memory systems can be loosely coupled with memory. Figure 3 - 2 shows two ways in which shared memory can be accessed: uniform memory access from the CPU on the left, and non-uniform memory access (NUMA) between the left and right disks.

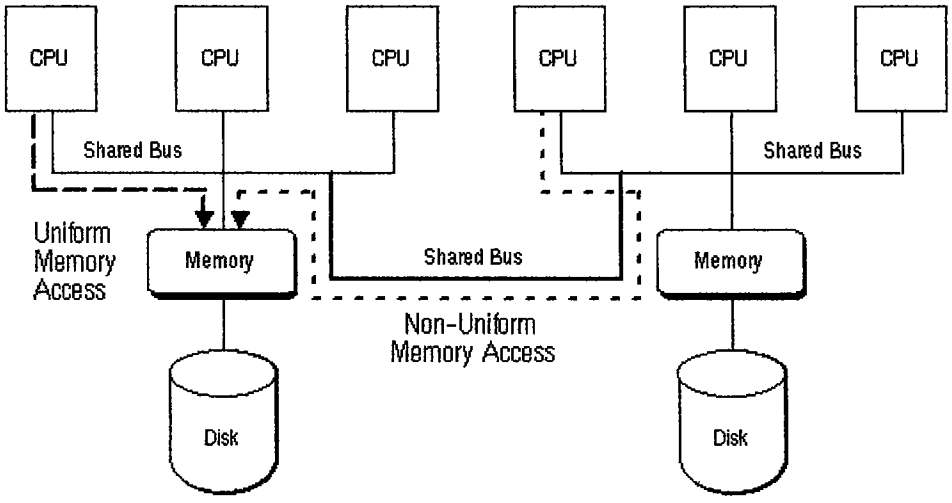

Figure 3 - 2. Uniform and Non-Uniform Memory Access

The Oracle Parallel Server can work with either form of memory access--but NUMA is a more expensive form of memory access and synchronization than uniform memory access. While any CPU can access the memory, it is more costly for the remote nodes to do this.

Summary: Shared Memory Systems

Parallel processing advantages of shared memory systems are described in this section.

Advantages

- Memory access is cheaper than inter-node communication. This means that internal synchronization is faster than using the distributed lock manager.
- Shared memory systems are easier to administer than a cluster.

Shared Disk Systems

Shared disk systems are typically loosely coupled. This section describes:

Loosely Coupled Systems

Loosely coupled shared disk systems, illustrated in Figure 3 - 3, have the following characteristics:

- Each node consists of one or more CPUs and associated memory.
- Memory is not shared between nodes.
- Communication occurs over a common high-speed bus.
- Each node has access to the same disks and other resources.
- A node can be an SMP if the hardware supports it. For example, nCUBE does not support an SMP, but many loosely coupled UNIX systems do support SMPs.
- Bandwidth of the high-speed bus limits the number of nodes (scalability) of the system.

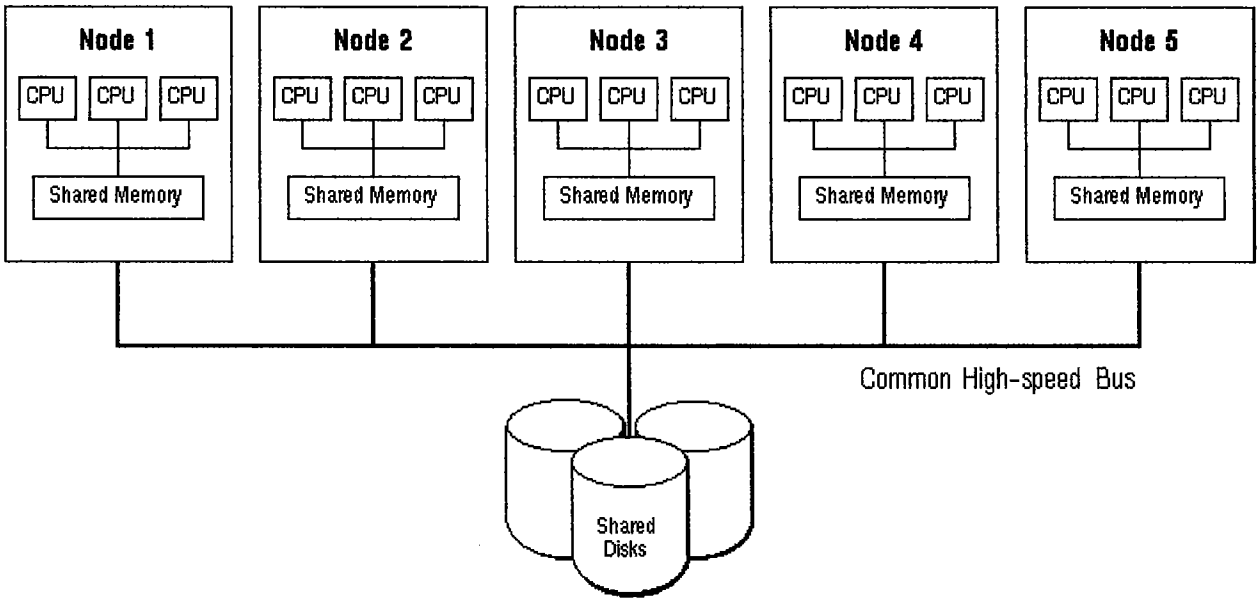

Figure 3 - 3. Loosely Coupled Shared Disk System

The cluster illustrated in Figure 3 - 3 is composed of multiple tightly coupled nodes. A DLM is required. Examples of loosely coupled systems are VAXclusters or Sun clusters.

Since the memory is not shared among the nodes, each node has its own data cache. Cache consistency must be maintained across the nodes and a lock manager is needed to maintain the consistency. Additionally, instance locks using the DLM on the Oracle level must be maintained to ensure that all nodes in the cluster see identical data.

There is additional overhead in maintaining the locks and ensuring that the data caches are consistent. The performance impact is dependent on the hardware and software components, such as the bandwidth of the high-speed bus through which the nodes communicate, and DLM performance.

Summary: Shared Disk Systems

Parallel processing advantages and disadvantages of shared disk systems are described in this section.

Advantages

- Shared disk systems permit high availability. All data is accessible even if one node dies.
- These systems have the concept of one database, which is an advantage over shared nothing systems.
- Shared disk systems provide for incremental growth.

Disadvantages

- Inter-node synchronization is required, involving DLM overhead and greater dependency on high-speed interconnect.
- If the workload is not partitioned well, there may be high synchronization overhead.
- There is operating system overhead of running shared disk software.

Shared Nothing Systems

Shared nothing systems are typically loosely coupled. This section describes:

- Overview of Shared Nothing Systems
- Massively Parallel Systems
- Summary: Shared Nothing Systems

Overview of Shared Nothing Systems

In shared nothing systems only one CPU is connected to a given disk. If a table or database is located on that disk, access depends entirely on the CPU which owns it. If the CPU fails the data cannot be accessed--regardless of how many other CPUs may still be running. Shared nothing systems can be represented as follows:

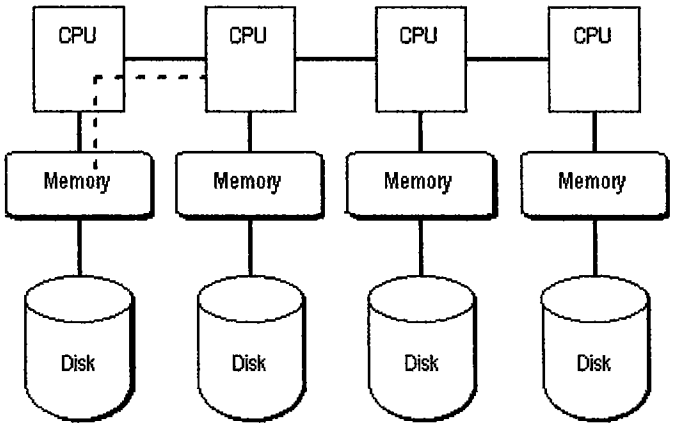

Figure 3 - 4. Shared Nothing System

Shared nothing systems are concerned with access to disks, not access to memory. Oracle Parallel Server can access the disks on a shared nothing system as long as the operating system provides transparent disk access, but this access is expensive in terms of latency.

Shared nothing systems are fundamentally concerned with access to disks, not memory. Nonetheless, adding more CPUs and disks can improve scaleup.

Massively Parallel Systems

Massively parallel (MPP) systems, illustrated in Figure 3 - 5, have the following characteristics:

- From only a few nodes, up to thousands of nodes are supported.
- The cost per processor may be extremely low because each node is an inexpensive processor.
- Each node has associated non-shared memory.
- Each node has its own devices, but in case of failure other nodes can access the devices of the failed node (on most systems).
- Nodes are organized in a grid, mesh, or hypercube arrangement.
- Oracle instances can potentially reside on any or all nodes.

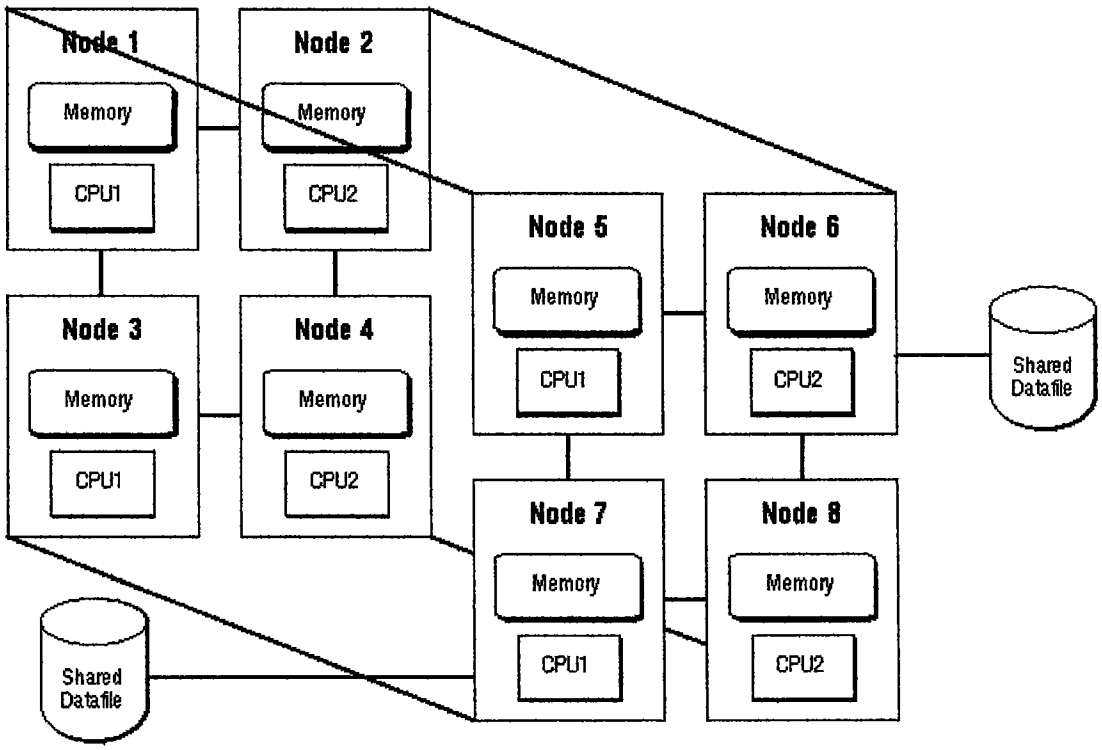

Figure 3 - 5. Massively Parallel System: A Hypercube Example

Note: A hypercube is an arrangement of processors such that each processor is connected to $\log_2 n$ other processors, where n is the number of processors in the hypercube. $\text{Log}_2 n$ is said to be the "dimension" of the hypercube. For example, in the 8-processor hypercube shown in Figure 3 - 5, dimension = 3; each processor is connected to three other processors.

A massively parallel system may have as many as several thousand nodes. Each node may have its own Oracle instance, with all the standard facilities of an instance. (An Oracle instance comprises the System Global Area and all the background processes.)

An MPP has access to a huge amount of real memory for all database operations (such as sorts or the buffer cache), since each node has its own associated memory. To avoid disk I/O, this advantage will be significant in long running queries and sorts. This is not possible for 32 bit machines which have a 2 GB addressing limit; the total amount of memory on an MPP system may well be over 2 GB.

As with loosely coupled systems, cache consistency on MPPs must still be maintained across all nodes in the system. Thus, the overhead for cache management is still present.

Examples of massively parallel systems are the nCUBE2 Scalar Supercomputer, the Unisys OPUS, Amdahl, Meiko, and the IBM SP.

Summary: Shared Nothing Systems

Parallel processing advantages and disadvantages of shared nothing systems are described in this section.

Advantages

- Shared nothing systems provide for incremental growth.
- System growth is practically unlimited.

- MPPs are good for read-only databases and decision support applications.
- Failure is local: if one node fails, the others stay up.

Disadvantages

- More coordination is required.
- A process can only work on the node that owns the desired disk.
- If one node dies, processes cannot access its data.
- Physically separate databases which are logically one database can be extremely complex and time-consuming to administer.
- Adding nodes means reconfiguring and laying out data on disks.
- If there is a heavy workload of updates or inserts, as in an online transaction processing system, it may be worthwhile to consider data-dependent routing to alleviate contention.

Shared Nothing /Shared Disk Combined Systems

A combined system can be very advantageous--one which brings together the advantages of shared nothing and shared disk, while overcoming their respective limitations. Such a combined system can be represented as follows:

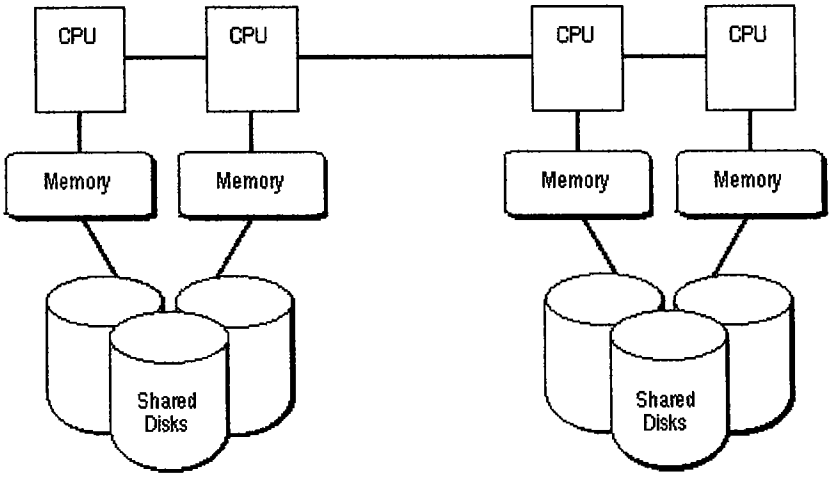

Figure 3 - 6. Two Shared Disk Systems Forming a Shared Nothing System

Here, two shared disk systems are linked to form a system with the same hardware redundancies as a shared nothing system. If one CPU fails, the other CPUs can still access all disks.
